“Network Parameter Outlier Algorithm for Graphical Identification of Biological Events“

Mark R. Bower

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Abstract (382 words)

Signal processing of data obtained from nervous systems commonly focuses on two types of signals: “states” of indeterminate duration, which are normally defined by a characteristic frequency (e.g., theta, gamma, sleep stages); and “events”, waveforms with a characteristic duration and shape (e.g., action potentials, sharp-wave/ripples, spindles, inter-ictal spikes). Analysis of events often centers on grouping or clustering events based on the assumption that waveforms that look similar arise from a single source (e.g., action potentials generated by an individual neuron). Historically, the simplest such technique was first to separate signal from noise using a threshold detector, which had the added benefit of reducing the amount of data stored to hard disks, which historically were expensive. Second, features (e.g., peak voltage, spikewidth) were computed for each event and then traditional clustering algorithms were applied to those features. In general, clustering techniques rely on iterative optimization methods that, while powerful, have time complexity that are, at best, $O(nlog(n))$, but are more commonly $O(n^2)$. An additional and often overlooked complication is that the shape and duration of biological signals are not statistically stationary; they may change over time or in response to fluctuating environmental variables (e.g., temperature, vigilance state, movement of electrodes). Signals related to the study of epilepsy are known to change over time [Bower et al., 2015]. The waveforms of inter-ictal spikes (IIS) have been shown to change following seizures [Bower et al., 2017], making clustering of IIS waveforms problematic for waveform-feature-based clustering algorithms recorded over multiple seizures.

Dynamic Graphical Systems (DGS) have a rich literature in regards to outlier detection, but not in regards to clustering. This is surprising, given the rich literature regarding unspurvised clustering in graphs, in general. Graphical clustering algorithms partition graphs into “neighborhoods” according to the connectivity patterns between groups of nodes, rather than pairwise comparisons of similarities between pairs of nodes.

Here, we present an algorithm based on sequential data analysis and unsupervised clustering based on wave shape correlations between neighboring signals, analogous to how human observers cluster signals when observing those signals in real time. The Network Parameter Outlier (NPO) algorithm sequentially places all peaks into a graph connected to contemporary peaks via edges whose weights are their respective correlation coefficients. Clustering is performed by graphical community detection algorithms being applied to a moving window of data. Importantly, this algorithm runs in O(n) time, allowing the anlaysis of datasets of any size as well as allowing real-time processing. We compare the performance of this approach to both fully-automated and semi-automated (the current gold standard) detection and clustering algorithms and show similar performance for stationary signals and improved performance by NPO for non-stationary signals.

Introduction

The analysis of biological data often involves the identification of repeated, time-bounded events (e.g., animal calls, action potentials, heartbeats). Historically, biological-signal processing algorithms have emphasized noise reduction (normally through spectral filtering and threshold/template detection), because such signals are temporally rare compared to background noise and because of the high costs that were associated with data storage. Over the past decades, however, data storage costs have decreased, dramatically, allowing "broadband" data acquisition from biological systems (i.e., continuous recording at high frequency). Despite this, noise reduction has remained the initial stage of most applications, ignoring the rich information contained in the acquired data that are not of interest (i.e., noise). One use for noise is outlier detection [Outliers]. In electrophysiology, outlier detection has been used successfully to identify states and events that are considered unimportant or distracting (e.g., artifact arising from subject movement, electrical stimulation) as well as significant states (e.g., seizures) by computing, for example, RMS voltage values in moving windows and testing for departures from baseline. Viewing biological events as outliers, however, has received much less attention.

Biological events consist of time-bounded events that share three properties that differ from noise events: signals are **rare** compared to noise, **repeatable** as they display a charaacteristic shape and **reliable** in frequency of occurrence, neither too fast or too slow. Biological signal detection algorithms normally are based on a pipeline algorithm: spectral filtering to emphasize the desired signal, thresholding to reduce noise detections, feature detection in the isolated waveforms and iterative optimization methods (e.g., Maximum Likelihood Estimation) to identify and cluster similar signals. This process has arisen through experience identifying specific signals of interest. Human observers, however, identify events of interest (i.e., "signals") even in data with which the observer may have little or no experience, applying "top-down" experience of how *all* signals *should* behave. Respiration, heartbeats, epileptiform events and action potentials all consist of a series of characteristically shaped, temporally-bounded signals separated by much longer periods of inactivity that are identifiable by human observers regardless of time- or amplitude-scale or generating mechanism (Figure 1).

Humans cannot visualize more than three dimensions, while computers can easily compute on hundreds of dimensions, but a similar difference exists in the time domain: computers can compare signals generated hours apart, while humans process data sequentially. Several classes of algorithms have been designed to mimic various aspects of human problem solving including Bayesian, Causality, Dempster-Shafer and AI approaches. These approaches commonly use graphical representations, because graphical representations of information possess many unique and useful properties. Among these properties is an inherent emphasis on the sequential nature of data.

Combining these ideas with graph-based community detection that uses temporal-graphical properties of biological events in addition to traditional time-voltage characteristics produced a novel signal identification algorithm; the Noise Outlier Graph-Based ("NOGB") algorithm. NOGB can identify any events that satisfy the three criteria. The free parameters are the expected duration of the event, the minimum acceptable amplitude and frequency bandwidth for filtering. NOGB computational time scales linearly with data size, runs in real-time, and adapts to changes in recording fidelity.

In addition, however, recordings of electrical biological signals are known to change over hours/days (e.g., due to changes in electrode position ("drift") [old paper showing changes in recorded action potential shape over days] or changes in ion concentrations around the electrode [action potential shape during seizures?]). In particular, the shape of IIS are known to change prior to seizures and following post-seizure sleep, reducing the reliability of signal detection algorithms that assume time-invariance in IIS shape.

Improvements in technology, however, have reduced both of these barriers. Readily-available computers can now record terabytes of data, eliminating the need for noise reduction, and parallel processing on computer clusters has given rise to bootstrapping, Deep Learning and highly-parallel techniques that no longer rely on an assumption of analytical statistical distributions (e.g., bootstrapping). Recent advances in mathematical graph theory have provided grapical clustering algorithms (i.e., "commnunity identification") that scale linearly with the amount of data. These advances, however, have found limited application in bioligical signal processing.


The NPO algorithm was applied to three different datasets to identify clusters of similar signal events across multiple time scales: action potentials in rats (1 msec duration), sharp-wave/ripples in rats and patients (100 msec), and inter-ictal spikes in patients (200 msec). In addition to replicating prior results, we extended Seizure-Related Consolidation (SRC) theory by showing: 1) the necessity of post-seizure sleep for consolidation and 2) that the flexibility offered by the NOGB algorithm provides a novel algorithm for seizure prediction.

Whereas it is more likely to have problems handling larger datasets (i.e., to have problems of "scaling"), the NPO algorithm works better as the amount of data increases. Because the computational complexity (time) scales linearly with the amount of data, the NPO algorithm is best-suited to handle the largest datasets, taking advantage of the "Unreasonableness of Effectiveness of Data" (Norvig, 2009).

The advent of multiprocessor computation has placed "Big Data" capabilities in the hands of individual researchers. This has allowed for the emergence of algorithms that replace fixed assumptions about data and distributions with computationally intensive approches that rely solely on the data themselves, but few examples exist where these new capabilities have been utilized. So-called "bootstrap" algorithms are but one example [...]. Molecular biology and genetics, in particular, have adapted to these advances and no utilize state-of-the-art computational strategies. Systems neuroscience has not, with many fields utilizing algorithms developed decades ago.

The field of extracellular neural signal detection and clustering, in particular, has continued to rely on classical optimization approaches, such as Maximum Likelihood Estimation, which assume underlying models consisting of ideal, analytic distributions such as a mixture-of-gaussians. Importantly, the field continues to rely on noise elimination as an early processings step, often ignoring the significant amount of infomration that could be utilized. This is particularly important in the analysis of bioligcal data, because so much of what is recorded is noise. Consider a system that is detecting action potentials from three different neurons that each have an average rate of 2 Hz. The duration of an action potential roughly 1 msec, so on average 6 msec of action potential waveforms will be recorded per second, leaving 994 msec of noise. The signal constitutes less than 1% of the recorded data, so noise elimintation invalidates over 99% of the acquired data *as a first step* [human neuroprosthetics paper]! This is a significant loss of information, even if it is only noise.

In addition, clustering algorithms have routinely relied on error minimization approaches [Duda and Hart]. These approaches were optimized for the strengths of computers, but are not inherently intuitive to humans, as is shown by the current methods used to teach physicians and neuroscientists to identify events of interest in electroencephalographic (EEG) recordings (i.e., to "read" EEG). Normally, recordings are observed over minutes or even hours, looking for events that share four properties: temporally-restricted, repeatable, rare and realistic. For example, sharp-wave/ripple complexes in rodent recordings are identified as brief (~100 msec), rare (<1 Hz) events that produce a characteristic (i.e., repeatable) sound over audio amplifiers without saturating those amplifiers (i.e., phsyiologically realistic). Trained observers reject both far more frequent, lower-amplitude background "noise" and far less frequent "artifacts" that often is often so large in amplitude as to be non-physiological (Figure 1).

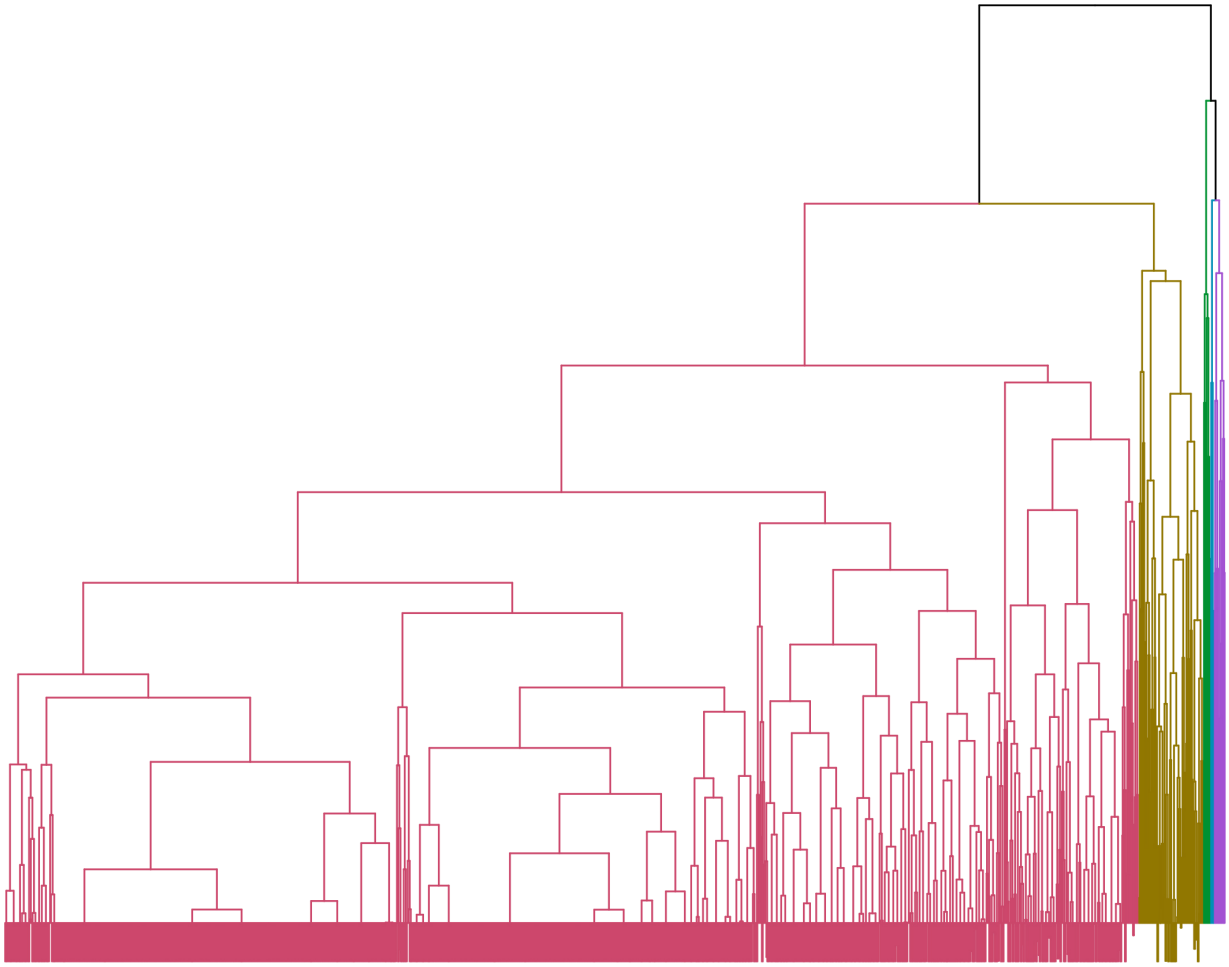
Other approaches have been less utilized. One such area is graphical/network methods that utilize connectivity to identify clusters (called "communities" in network analysis [Barabasi]. If the factors that govern connectivity are time-local, then clusters naturally become time-varying and thus les sensitive to time-varying factors such as drift.

The algorithm consists of three stages: 1) detect all peaks in a moving window, form a graph using correlation coefficient of time-voltage waveforms as edge weights and identify clusters using graphical algorithms, 2) reconstruct graphs for the entire recording duration for each cluster and compute graphical properties, 3) identify those clusters whose properties differ from that of the most common detections (i.e., "noise"). This algorithm requires four parameters based on expected physiological properties of events: 1) characteristic duration, 2) characteristic spectral window, 3) minimum average firing rate, and 4) maximum average firing rate. These properties (the last two, in particular) are often used after clustering to reject clusters that do not meet expected physiological parameters. The NOGB algorithm, instead, uses these physiological expectations at the beginning of the detection process to aid in clustering.

Is this any better than a simple threshold detector? How to compare? One advantage is that the identification step produces clusters that could be used in subsequent, standard cluster-joining steps.

The Iron Law of Noise: Low count. Small peak. All along, I have assumed that a 2D density was required to allow slopes or valleys to be identified as separatrices. Finally, it occurred to me that bootstrapping doesn't require any of that. Identify "noise" members using The Iron Law, then use bootstrapping to identify groups outside the 95% confidence noise interval. Simple and straight-forward.

Consolidation is a behavioral phenomenon. When we discuss the discuss the temporal decay of the effects of the preceding seizure, then, we will use the term "memory trace" or "trace" of the preceding seizure. This is not to suggest that seizures are learned, in the same manner as an event, an association or a behavioral task. Rather, it is intended to emphasize the underlying cellular plasticity mechanisms.



Halting on graph-related work to create a simple IIS detector for comparisons. I stopped at 'MySQL_analysisLoop_M2C('').R'.

Methods

Include a section on Unit Testing of your code. Show how it is used and how others can contribute.

Outline of Action

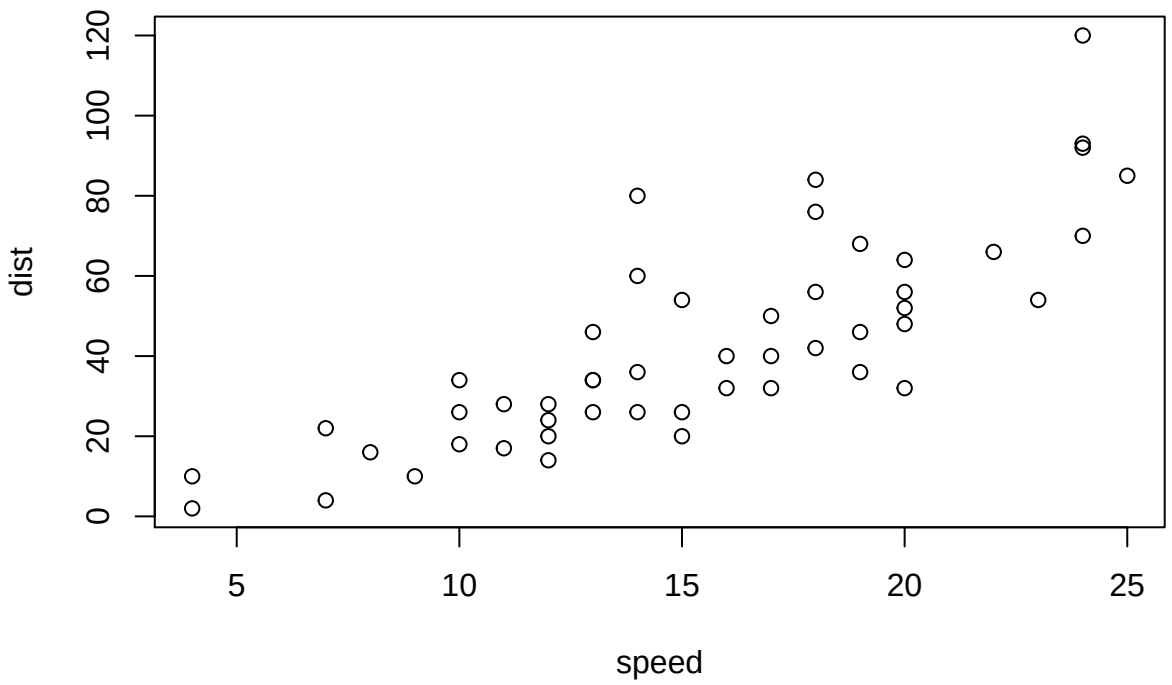
Make the Rmd document "live"; i.e., update and over-write it, continuously. It is not a journal or lab book. It is a living document that needs to be burned down, bulldozed and re-built on a daily basis. Posterity is for paper and pen. Get used to rewriting sections until they solidify over days. When a section achieves some stability, copy it into the Scrivener document.

Discussion

Define epileptoprospect (EPx) = T_SRCdecay / T_betweenSeizures. This suggests a quantitative answer to the question of whether "seizures beget seizures": it depends on whether the second seizure occurs quickly enough to be influenced by the decay of the preceding seizure and thus be more similar to (than would be expected by chance seizure dynamics) the first seizure. The question centers on which neurons and brain structures will participate in a given seizure. If the "memory trace" of the preceding seizure remains strong enough to influence the recruitment of neurons into the subsequent seizure, then the preceding seizure will indeed beget the subsequent seizure. This leads to a merging of terminology between epilepsy and learning. If the two phenomenon do indeed utilize similar cellular mechanisms, this merging is inevitable. While the "time between seizures" has always been a known and measurable quantity, we have supplied the "numerator" to this equation: the "time of trace decay".

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

```
plot(cars)
```



Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).